

Figure 1

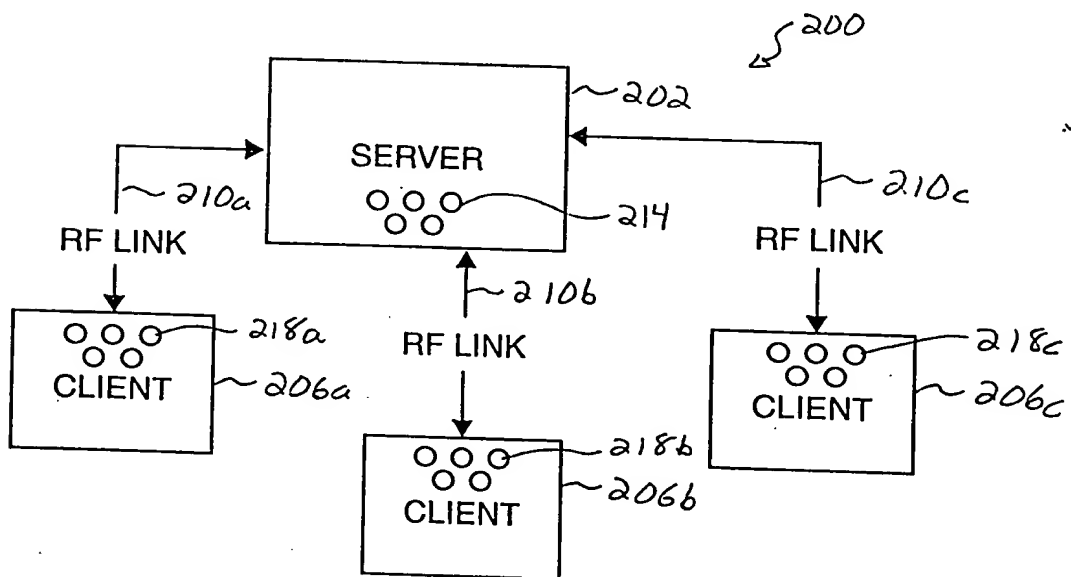


Figure 2a

250



Figure 26

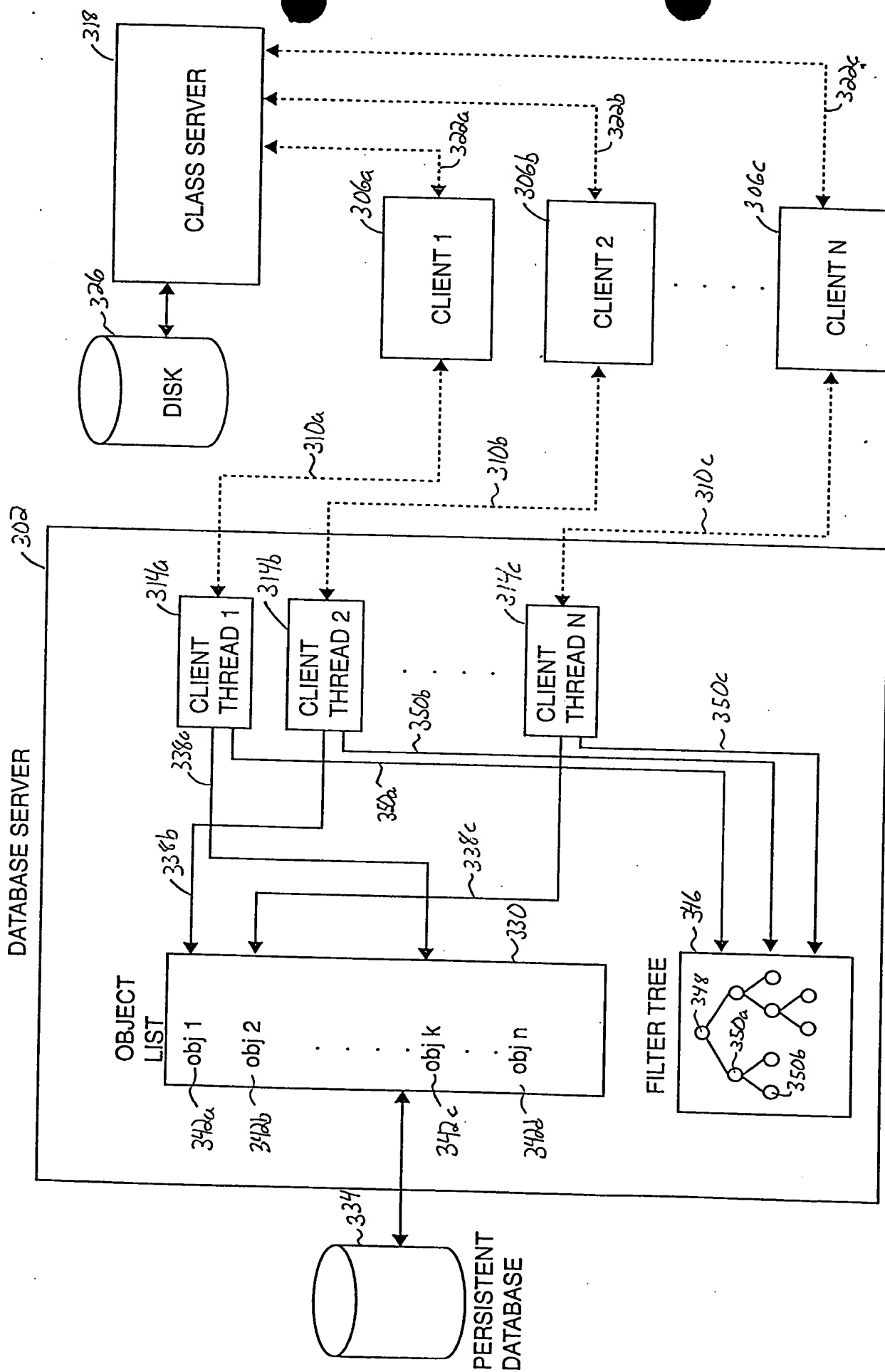


Figure 3

SITUATION DISPLAY

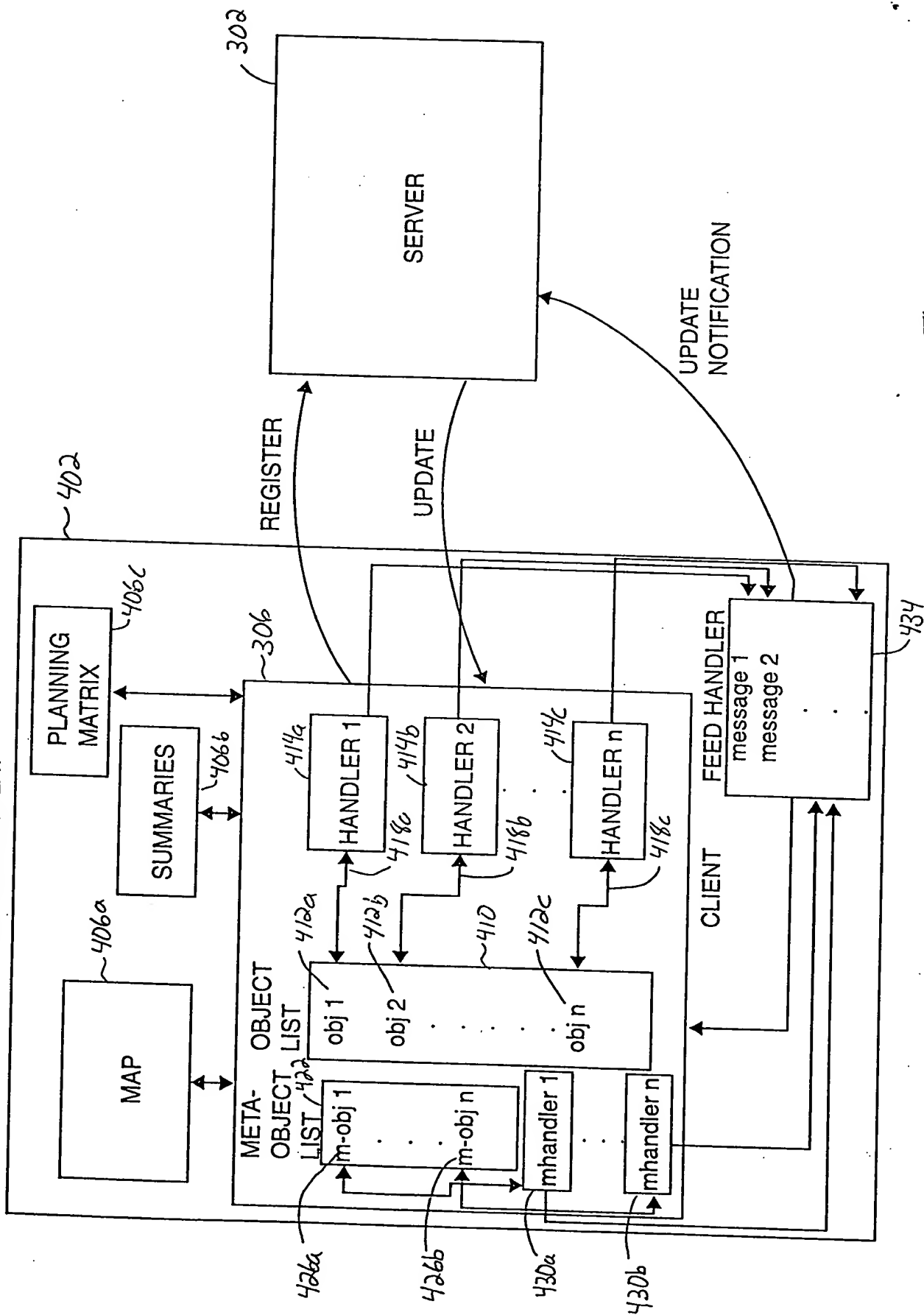


Figure 4

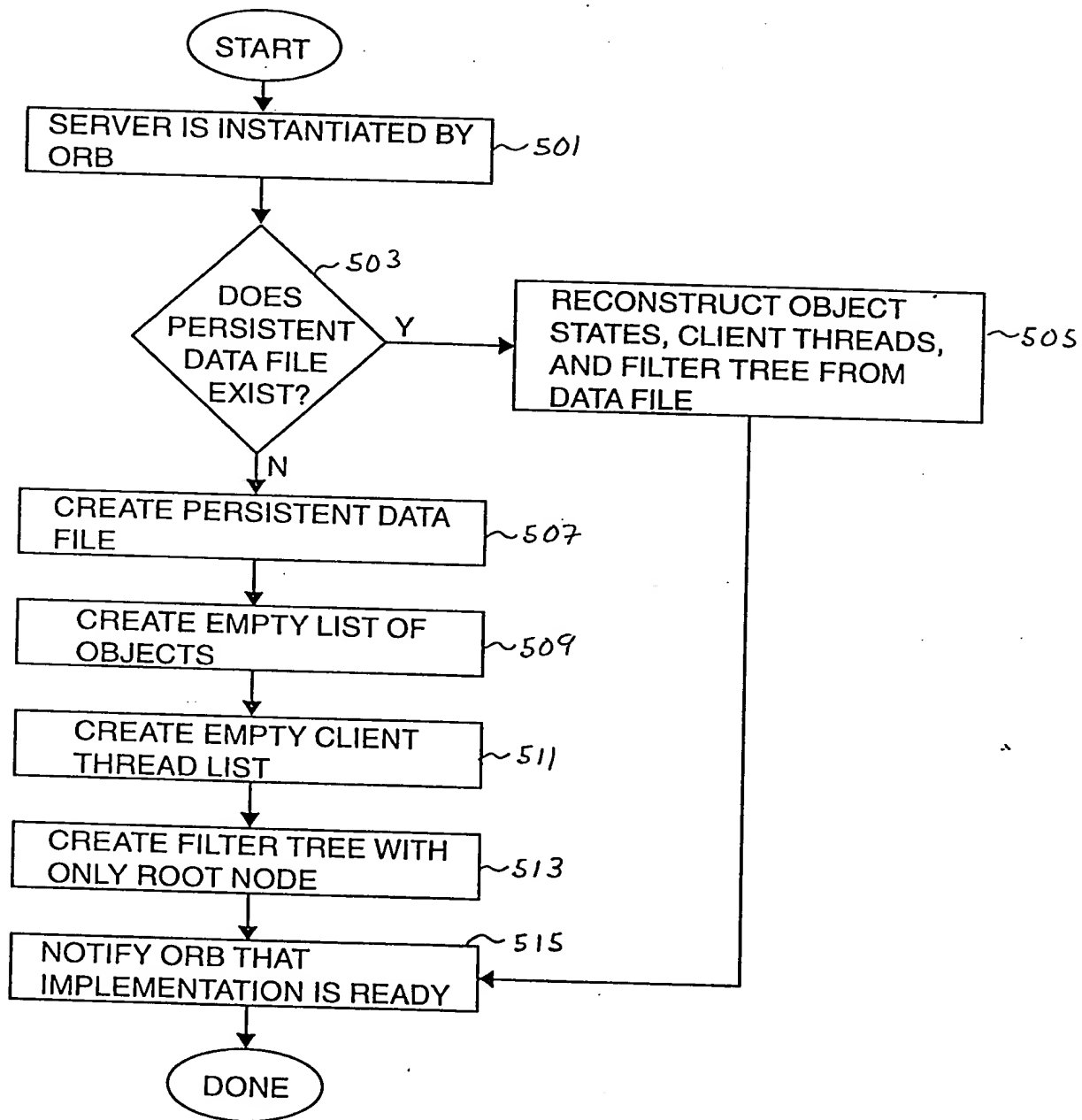


Figure 5

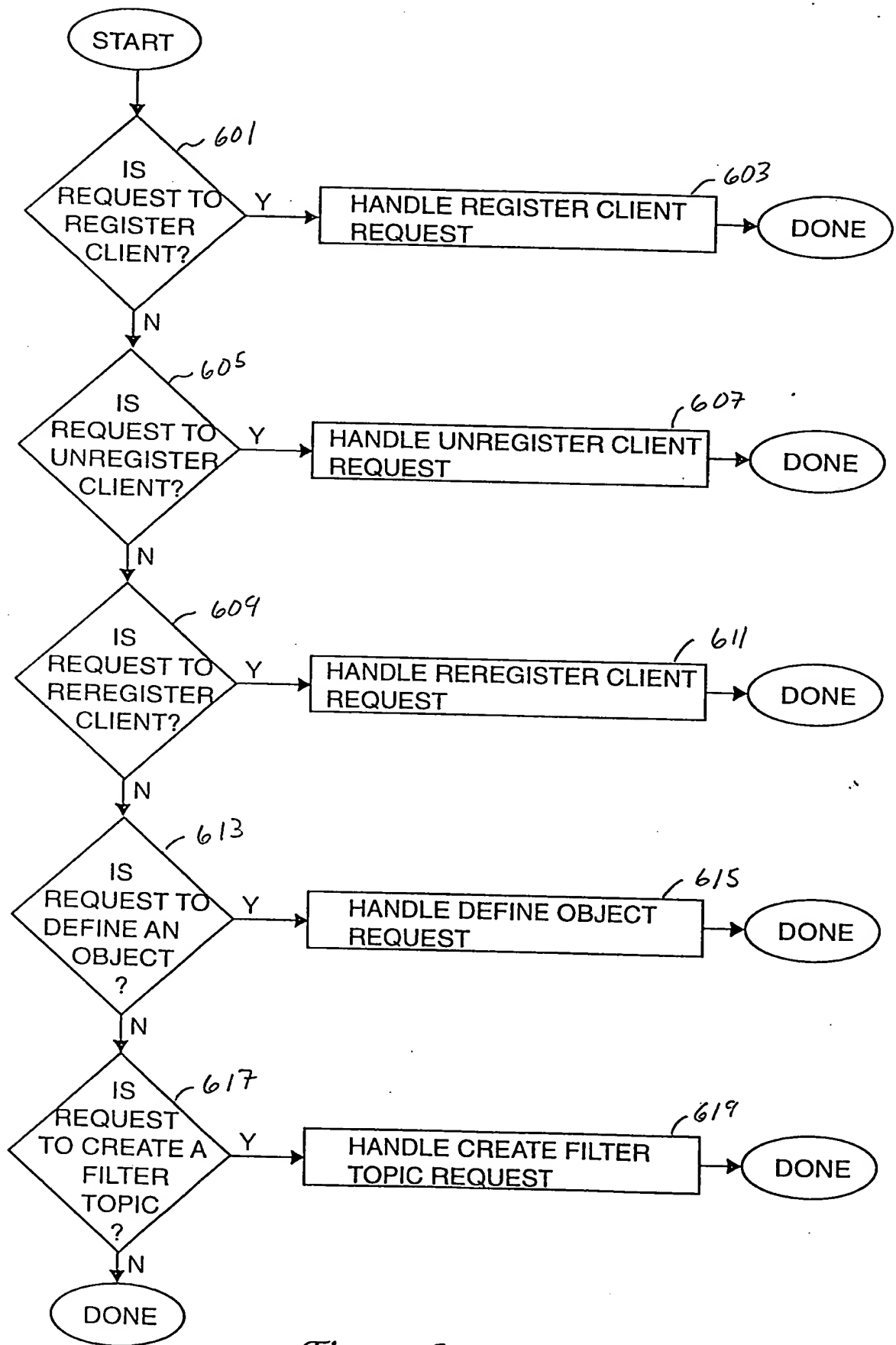


Figure 6

```

graph TD
    615((START 615)) --> 701[OBTAIN OBJECT LIST LOCK AND WRITE LOCK ON DATA FILE]
    701 --> 702{DOES AN UPDATE FOR SPECIFIED OBJECT ALREADY EXIST ON QUEUE ?}
    702 -- Y --> 704[REMOVE EXISTING REFERENCE TO UPDATED OBJECT FROM THE OBJECT LIST]
    702 -- N --> 703[PLACE OBJECT UPDATE AT END OF OBJECT LIST QUEUE]
    704 --> 703
    703 --> 705[COMMIT UPDATE TO DATA FILE]
    705 --> 707[RELEASE OBJECT LIST LOCK AND WRITE LOCK ON DATA FILE]
    707 --> 616((DONE))
  
```

Figure 7

```

graph TD
    603((START 603)) --> 801[GENERATE A UNIQUE TOKEN TO IDENTIFY CLIENT ~801]
    801 --> 803[OBTAIN A WRITE LOCK ON FILTER TREE AND OPEN TRANSACTION WITH DATA FILE ~803]
    803 --> 805[UPDATE FILTER TREE BASED ON PARAMETERS DEFINED BY CLIENT IN FILTER TOPIC LIST PASSED IN REGISTRATION REQUEST ~805]
    805 --> 807[COMMIT TRANSACTION ~807]
    807 --> 809[RELEASE LOCK ON FILTER TREE ~809]
    809 --> 811[OPEN SOCKET/PORT TO CLIENT ~811]
    811 --> 813[START CLIENT THREAD IN SERVER ~813]
    813 --> 815[RETURN UNIQUE TOKEN TO CLIENT ~815]
    815 --> DONE((DONE))
  
```

DONE

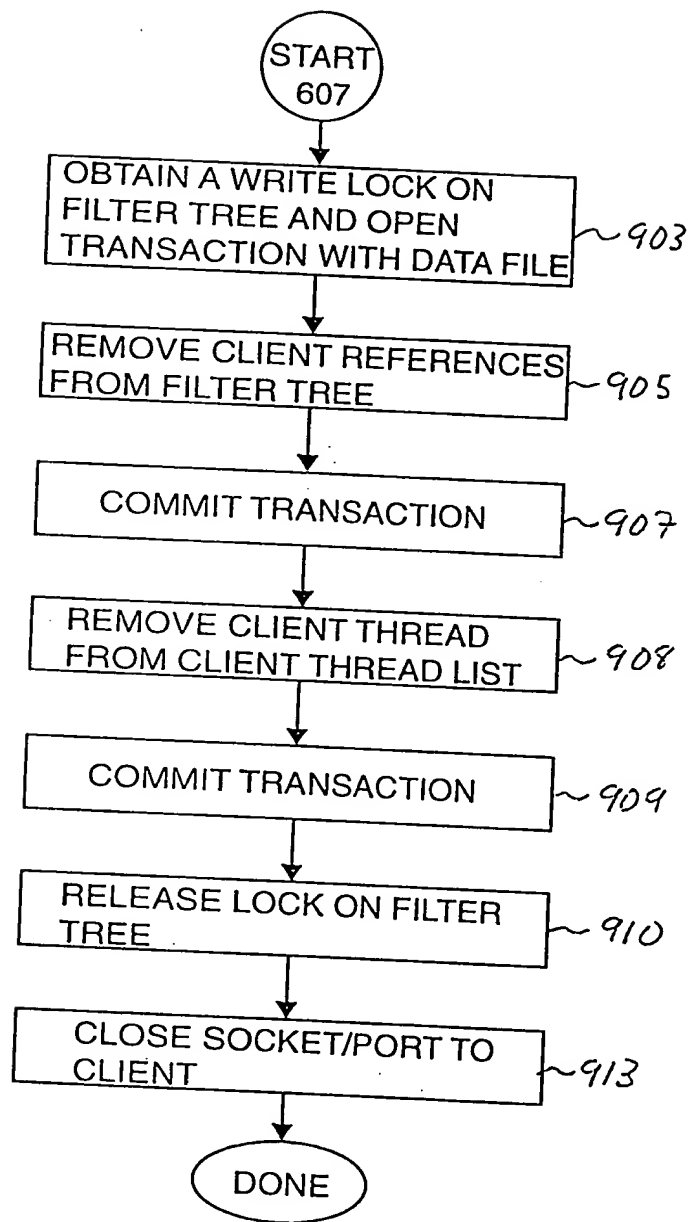
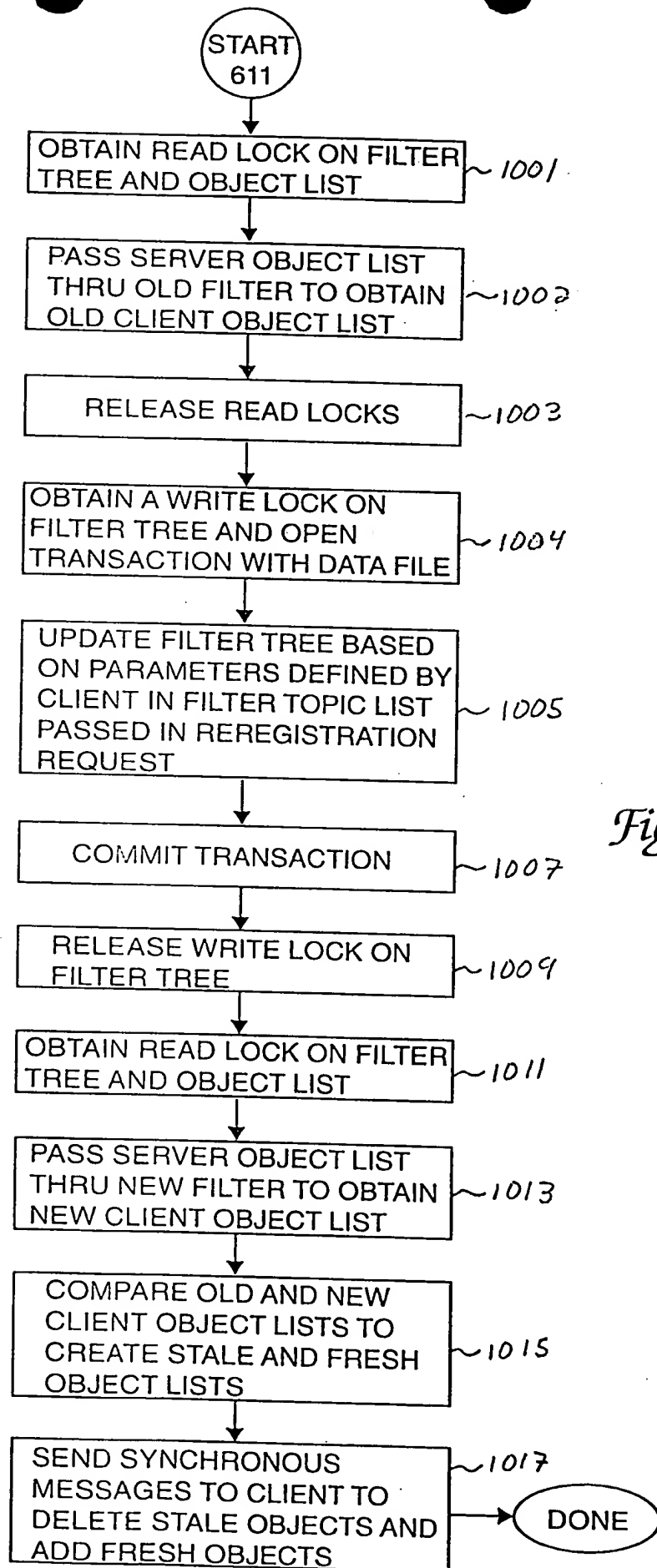


Figure 9



```
graph TD; 619((START 619)) --> 1103[OBTAIN A WRITE LOCK ON FILTER TREE AND OPEN WRITE TRANSACTION WITH DATA FILE 1103]; 1103 --> 1105[UPDATE FILTER TREE TO INCLUDE NEW TOPIC 1105]; 1105 --> 1107[COMMIT TRANSACTION 1107]; 1107 --> 1109[RELEASE LOCK ON FILTER TREE 1109]; 1109 --> DONE((DONE));
```

DONE

```

graph TD
    START((START 813)) --> 1202[OBTAIN READ LOCK ON OBJECT LIST]
    1202 --> 1204[SET CURRENTLY REFERENCED OBJECT]
    1204 --> 1207[RELEASE READ LOCK]
    1207 --> 1208{DOES OBJECT PASS FILTER?}
    1208 -- N --> 1217[ADVANCE TO NEXT OBJECT ON LIST]
    1208 -- Y --> 1211[SEND OBJECT TO CLIENT]
    1211 --> 1219{ACKNOWLEDGEMENT OF RECEIPT?}
    1219 -- N --> 1208
    1219 -- Y --> 1217
    1217 --> 1202

```

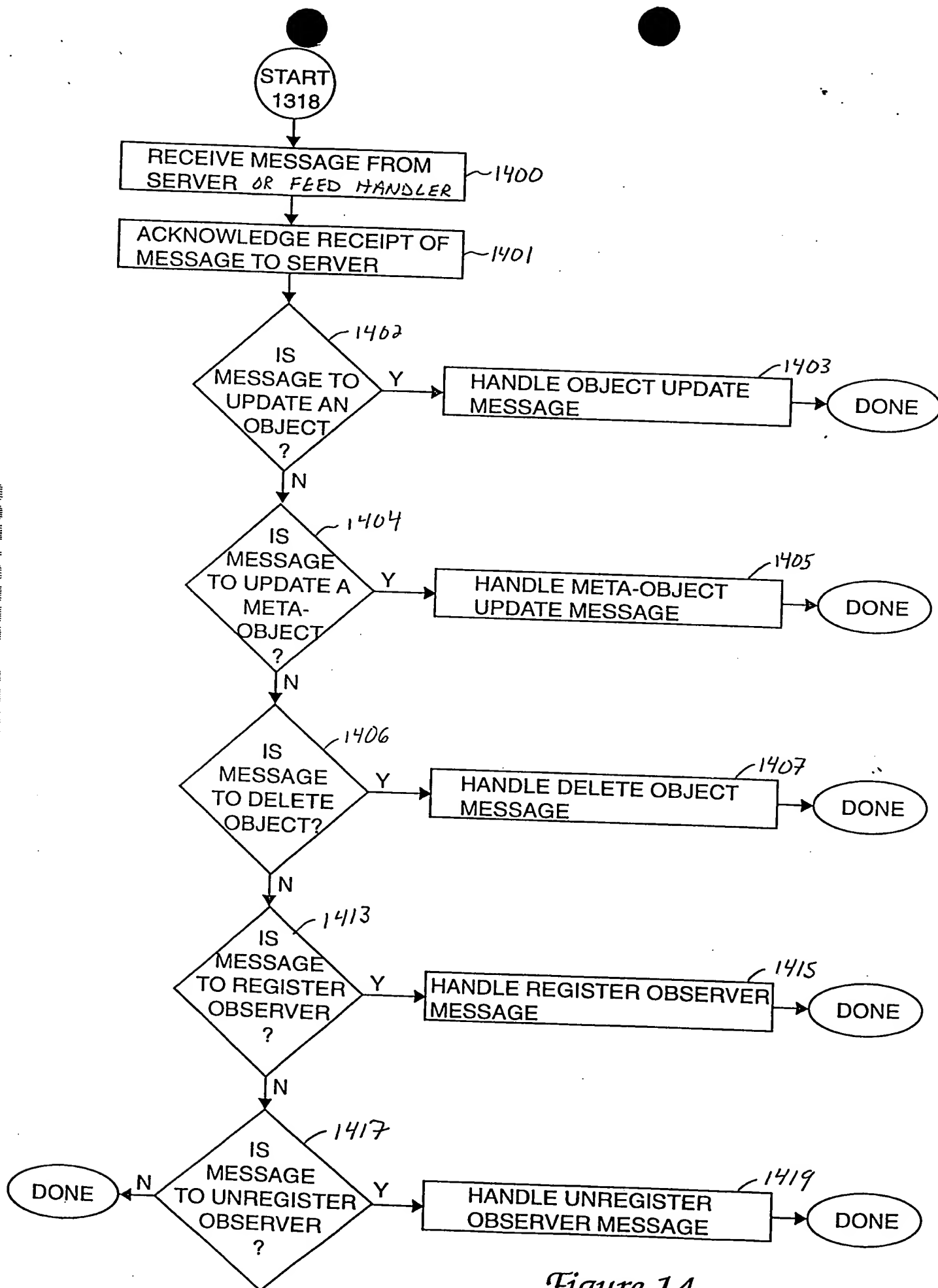
Figure 12

```

graph TD
    START([START]) --> 1301[OBTAIN USER INPUT]
    1301 --> 1302[CREATE DISPLAY WINDOWS AND LOAD IN GUI FILES FOR OBSERVERS]
    1302 --> 1304{IS CLIENT IN SOLO MODE?}
    1304 -- Y --> 1306{DOES PERSISTENT DATA FILE EXIST?}
    1304 -- N --> 1312[REGISTER CLIENT WITH ORB]
    1306 -- Y --> 1311[RECONSTRUCT OBJECT STATES]
    1306 -- N --> 1308[CREATE PERSISTENT DATA FILE]
    1308 --> 1310[CREATE EMPTY LIST OF OBJECTS]
    1311 --> 1310
    1310 --> 1318[LISTEN FOR UPDATES]
    1312 --> 1314[RECEIVE CLIENT TOKEN DIRECTLY FROM SERVER]
    1314 --> 1316[ESTABLISH SOCKET CONNECTION TO SERVER]
    1316 --> 1317[START FEED HANDLER]
    1317 --> 1318
    1318 --> 1318

```

Figure 13



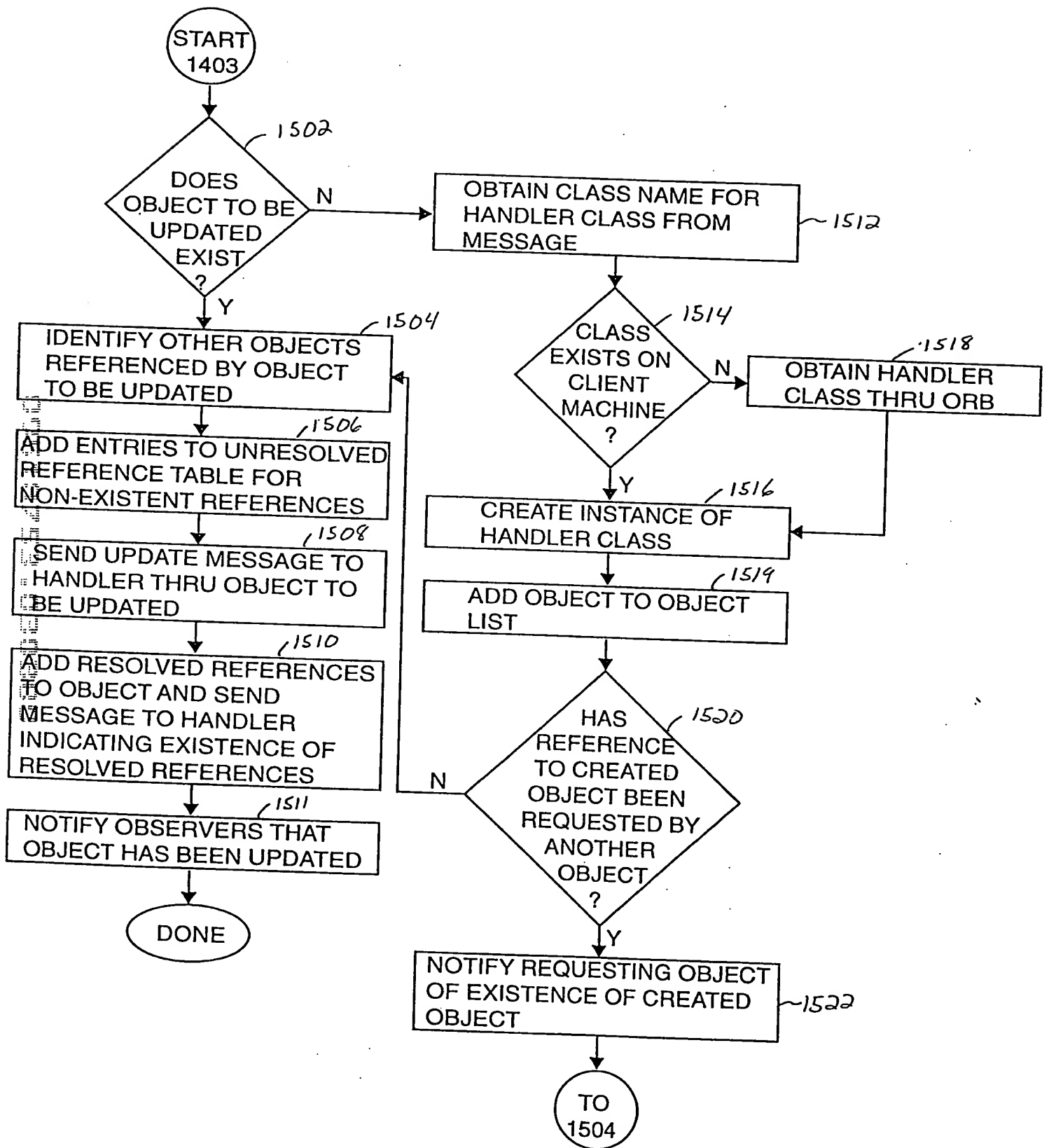


Figure 15

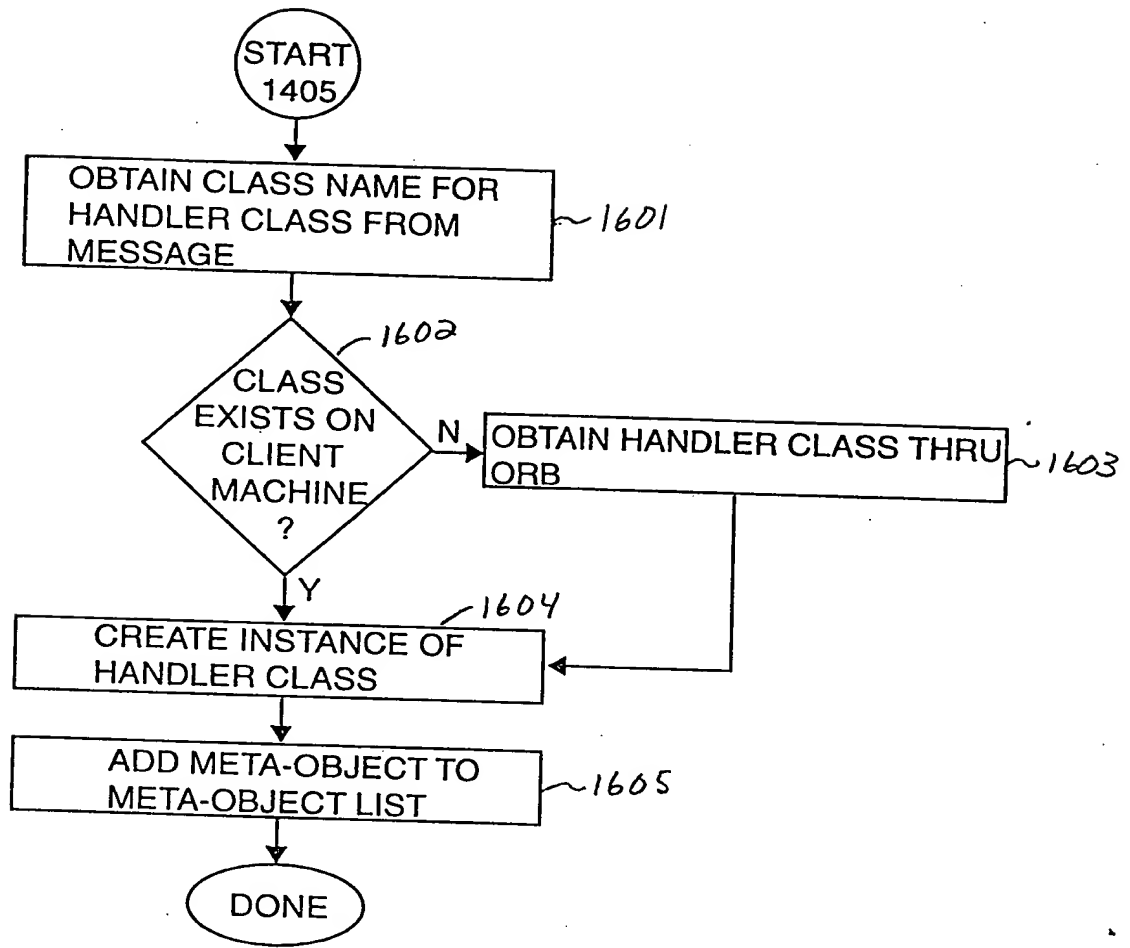


Figure 16

Figure 17

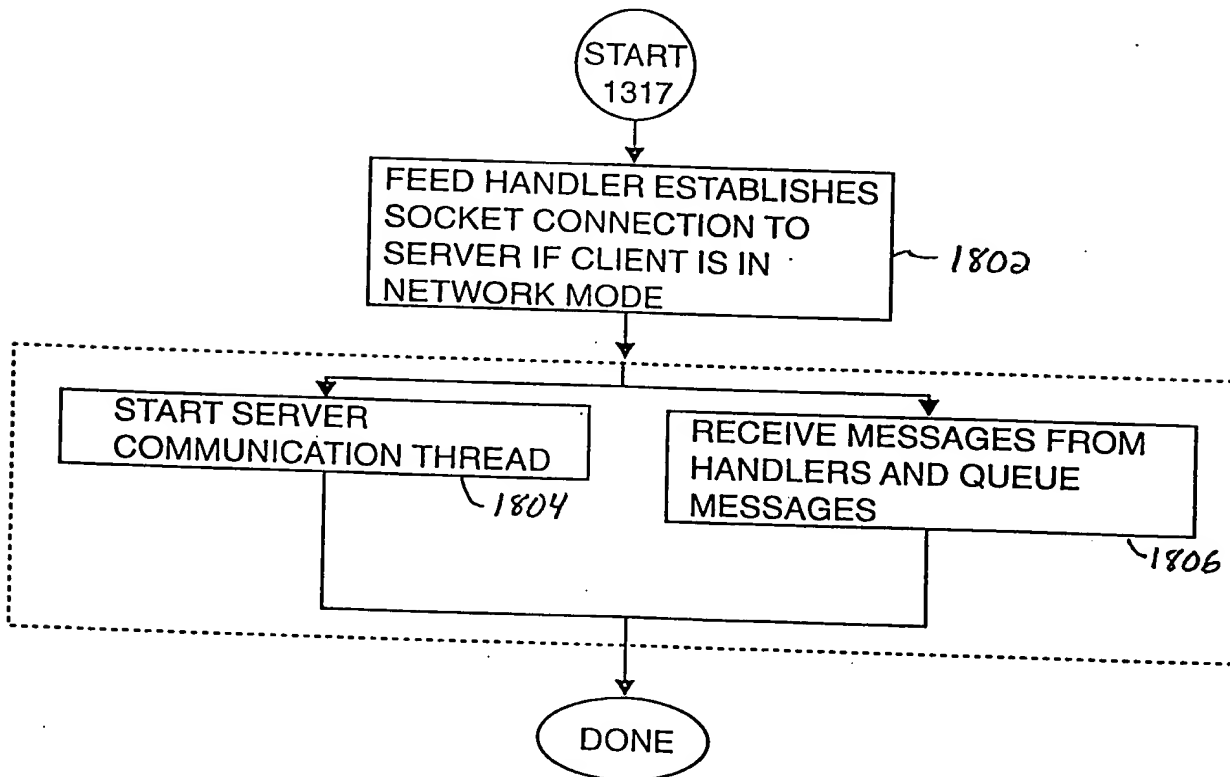


Figure 18

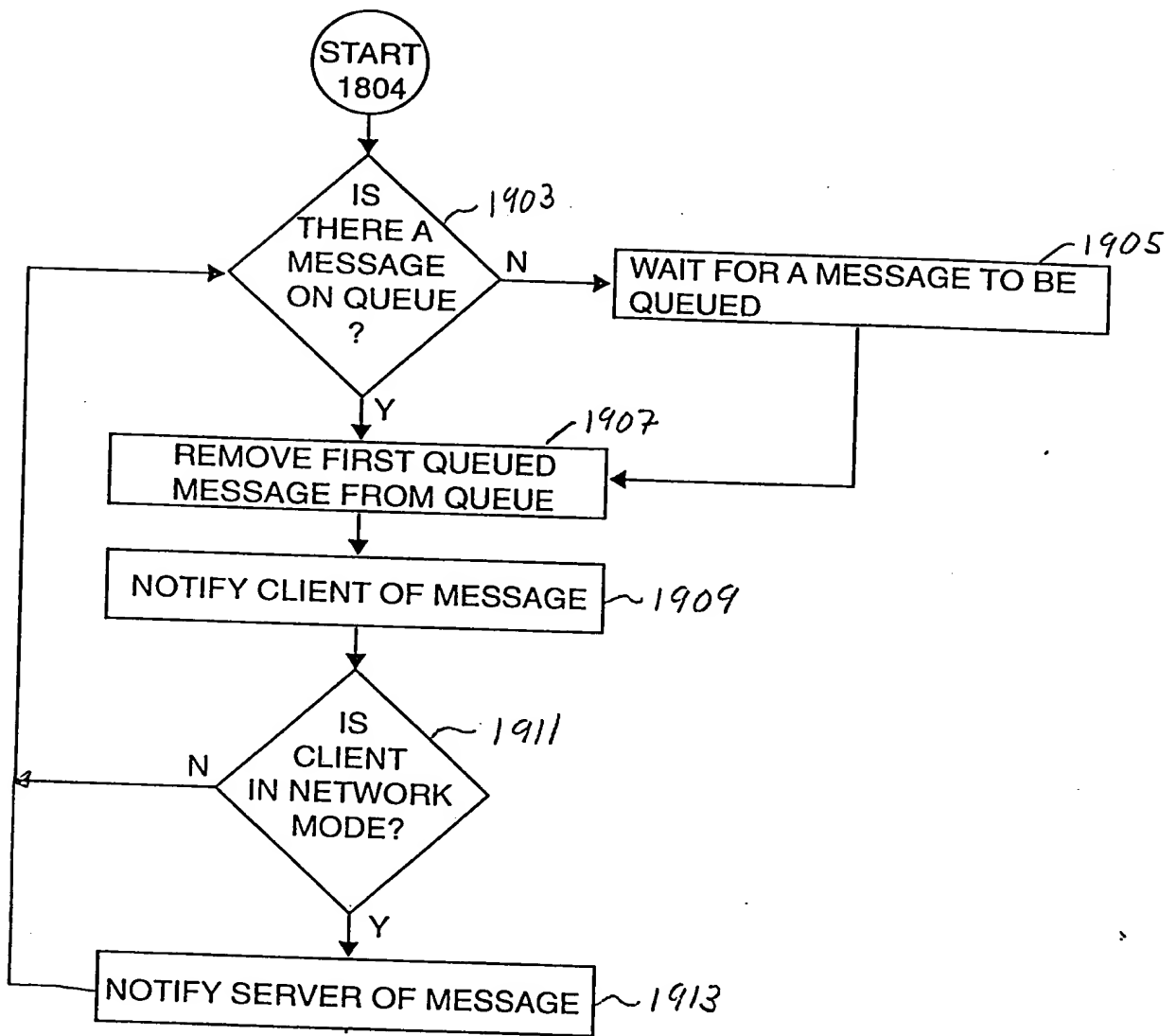


Figure 19

```
graph TD; 1516((START 1516)) --> 2002[CREATE GRAPHICAL COMPONENT FOR HANDLER]; 2002 --> 2004[REGISTER GRAPHICAL COMPONENT WITH SITUATION DISPLAY]; 2004 --> DONE([DONE]);
```

Flowchart 2000 illustrates the process of creating a graphical component for a handler and registering it with a situation display. The process begins with a start node (1516), followed by the step "CREATE GRAPHICAL COMPONENT FOR HANDLER" (2002), then "REGISTER GRAPHICAL COMPONENT WITH SITUATION DISPLAY" (2004), and finally ends at a "DONE" node.

Figure 20

```

graph TD
    Start([START  
1519]) --> 2101[OBTAIN WRITE LOCK FOR  
OBJECT LIST]
    2101 --> 2103[WRITE OBJECT TO OBJECT LIST]
    2103 --> 2105[RELEASE WRITE LOCK]
    2105 --> 2107{IS  
CLIENT IN  
SOLO MODE  
?}
    2107 -- N --> Done1([DONE])
    2107 -- Y --> 2109[OPEN TRANSACTION WITH  
PERSISTENT DATA FILE]
    2109 --> 2111[STORE UPDATE MESSAGE IN  
PERSISTENT DATA FILE]
    2111 --> 2113[COMMIT TRANSACTION]
    2113 --> Done2([DONE])
  
```

Figure 21

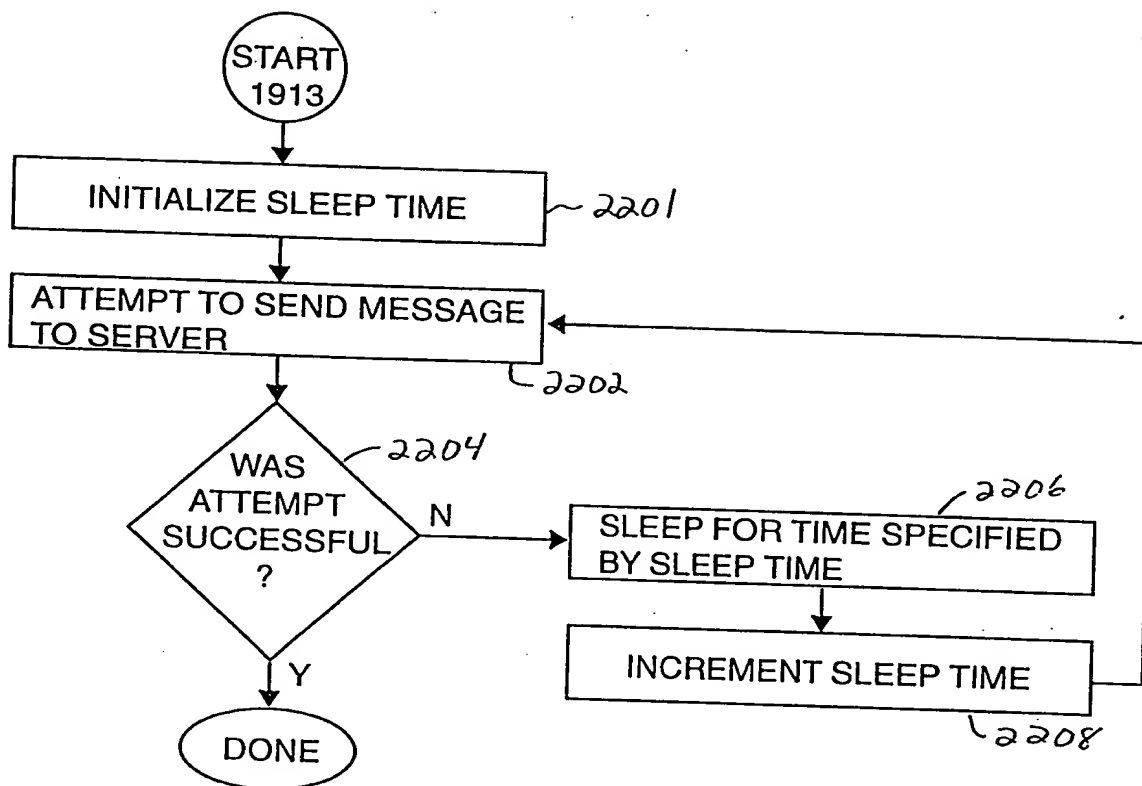


Figure 22

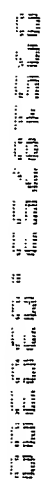


Figure 23

2025 RELEASE UNDER E.O. 14176

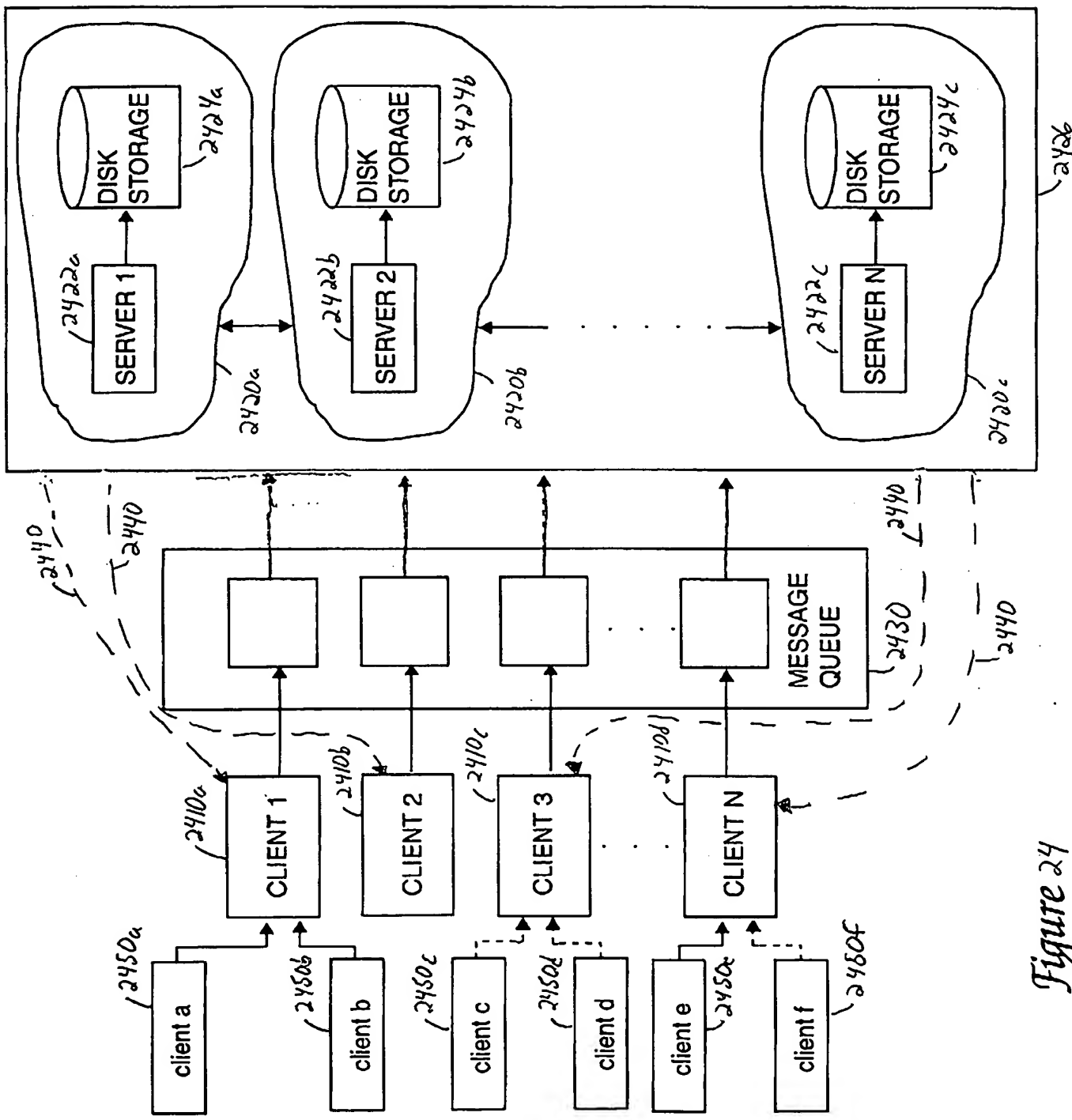
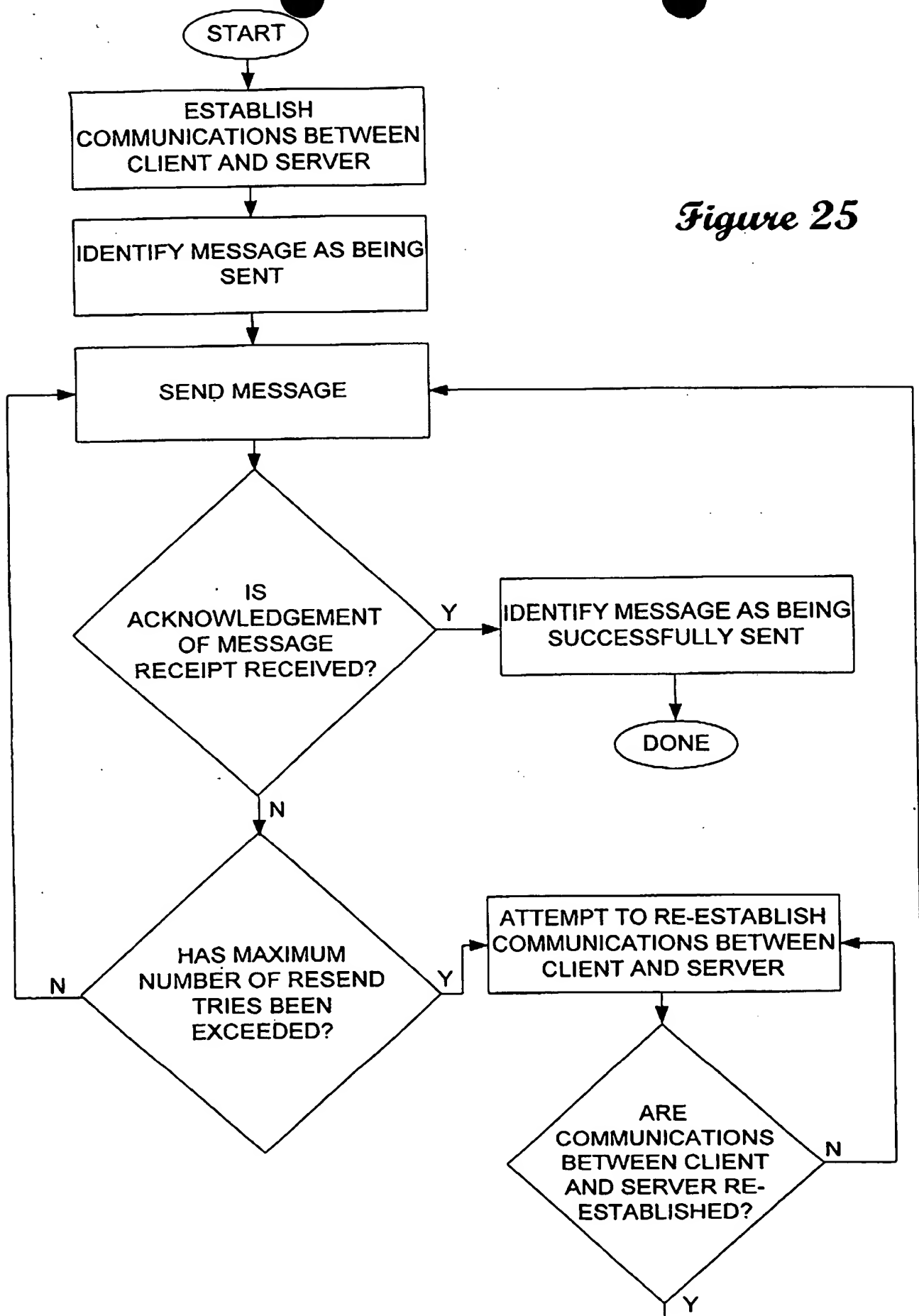


Figure 24



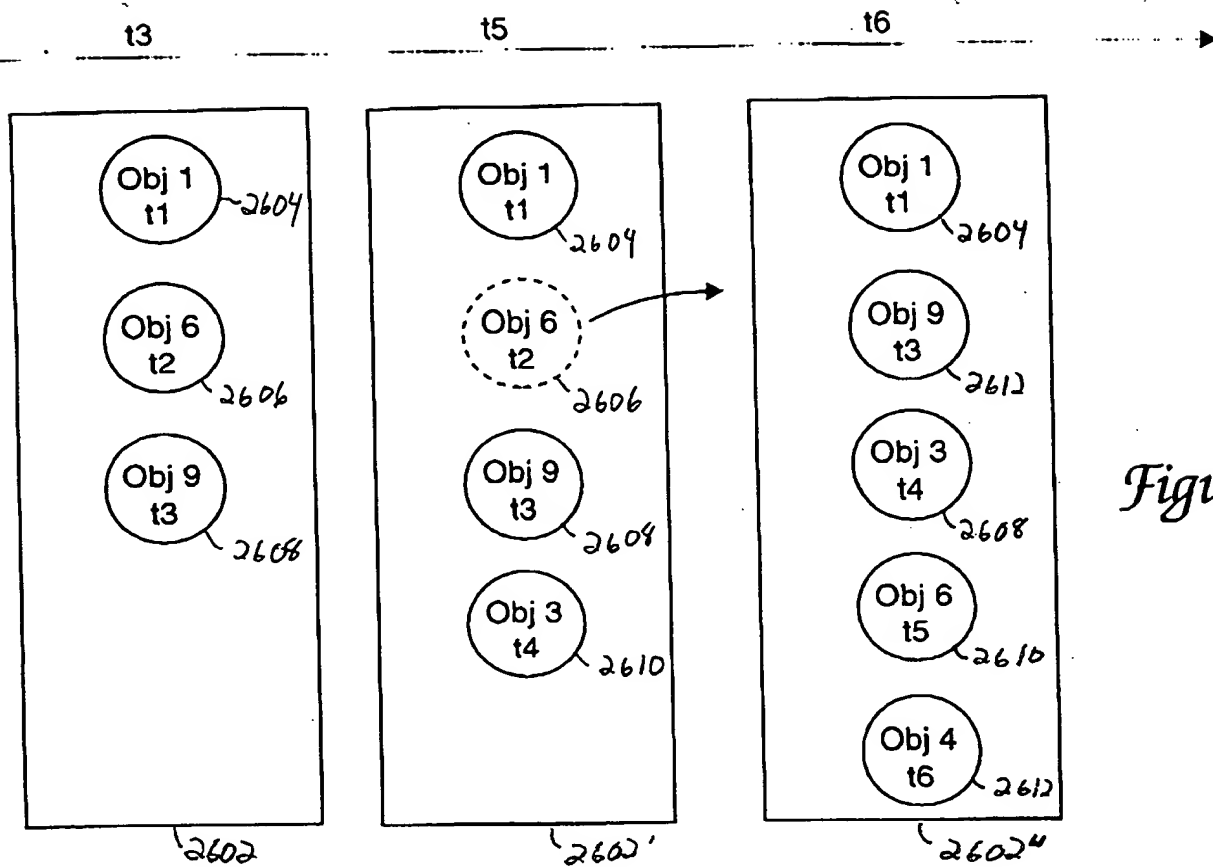


Figure 26a

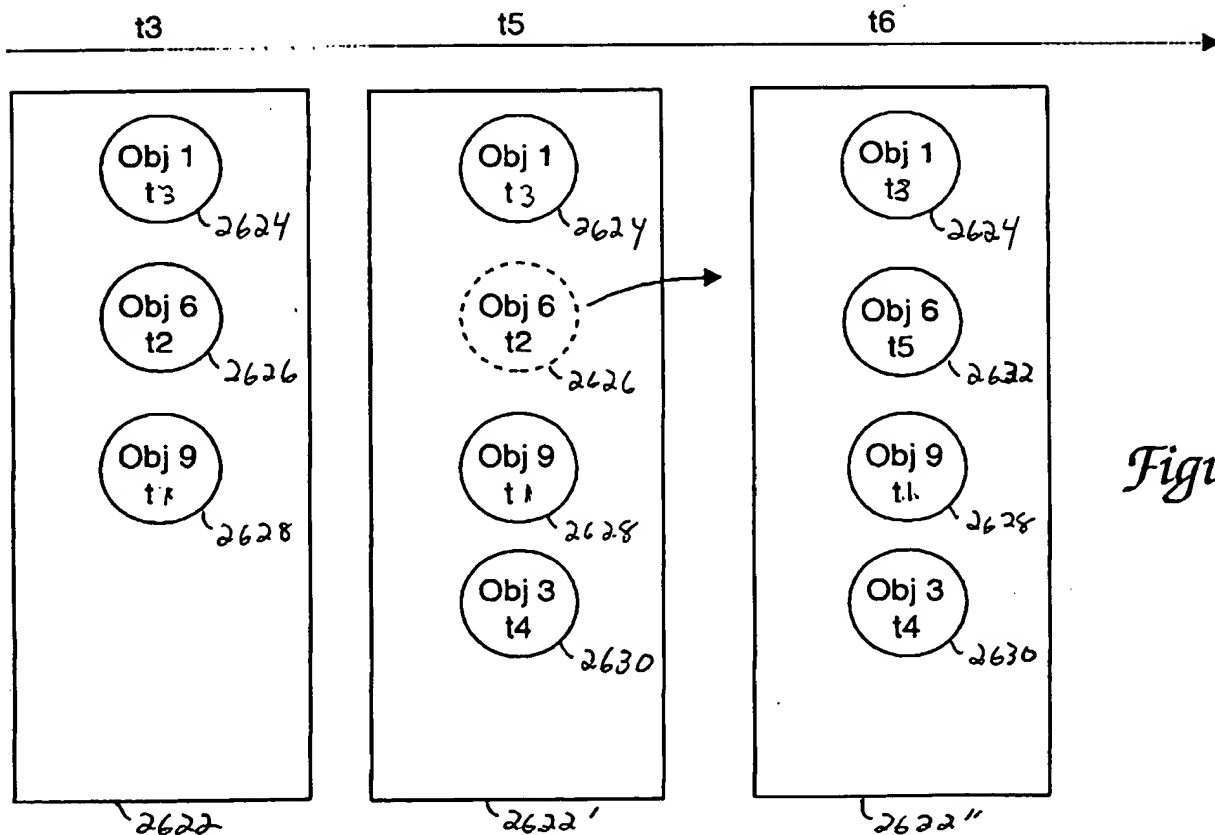


Figure 26b